# CELLS

## A virtual Smartphone Architecture

Jeremy Andrus  Christoffer Dall  Alexander Van't Hof  Oren Laadan  Jason Nieh

Columbia University
in the city of new york



23rd ACM Symposium on Operating Systems Principles, Cascais, Portugal, October 2011

# CELLS

## A virtual Smartphone Architecture

Jeremy Andrus  Christoffer Dall  Alexander Van't Hof  Oren Laadan  Jason Nieh

Columbia University
in the city of new york



23rd ACM Symposium on Operating Systems Principles, Cascais, Portugal, October 2011

# Personal Phone

# Business Phone


# Personal Phone

## Business Phone


## Personal Phone

## Developer Phone

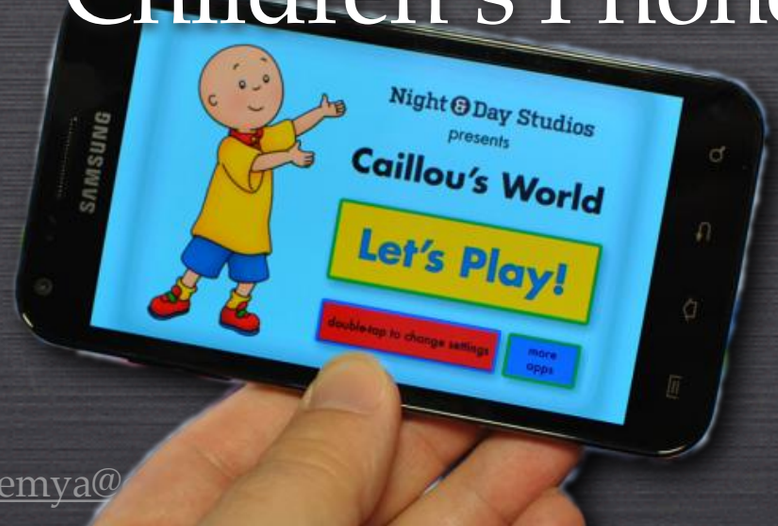

 

## Business Phone


## Personal Phone

## Developer Phone


## Children's Phone





# Virtualization

# Server Virtualization
## Bare-Metal Hypervisor

# Server Virtualization

## Bare-Metal Hypervisor

Hypervisor / VMM

Hardware

# Server Virtualization
## Bare-Metal Hypervisor

# Server Virtualization
## Bare-Metal Hypervisor

poor device support / sharing

# Desktop Virtualization

## Hosted Hypervisor

host user space

Hardware

# Desktop Virtualization

## Hosted Hypervisor

host user space


Hypervisor / VMM

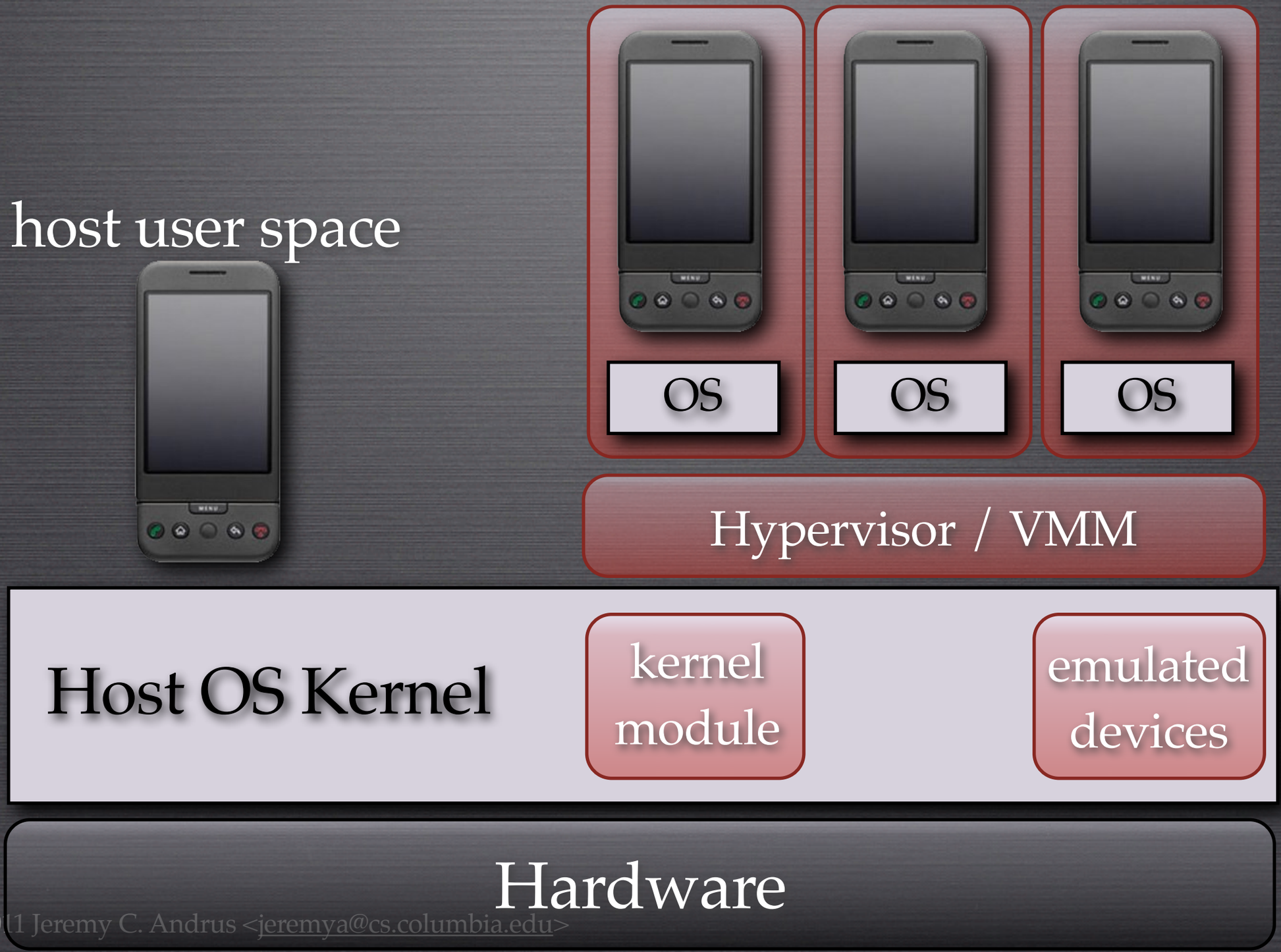
Host OS Kernel

kernel module

Hardware

# Desktop Virtualization

## Hosted Hypervisor



# Desktop Virtualization

## Hosted Hypervisor

poor device performance

host user space

OS

Hypervisor / VMM

Host OS Kernel

kernel module

emulated devices

Hardware

# Non-Virtualization
## User Space SDK

Hardware

# Non-Virtualization
## User Space SDK

# Non-Virtualization
## User Space SDK

no standard apps
less secure

# Non-Virtualization
## User Space SDK

no standard apps
less secure

custom user space API for isolated apps

OS Kernel

Hardware

# Key Challenges

# Key Challenges

- device diversity

# Key Challenges

- device diversity

WiFi

# Key Challenges

- device diversity

WiFi

Accelerometer

# Key Challenges

- device diversity

GPS

WiFi

Accelerometer

# Key Challenges

- device diversity

GPS

Cell Radio

WiFi

Accelerometer

# Key Challenges

- device diversity

GPS

Cell Radio

WiFi

Framebuffer

Accelerometer

# Key Challenges

- device diversity

| | | | |
|---|---|---|---|
| microphone | headset | | |
| Power | Touchscreen | Buttons | GPS |
| Cell Radio | WiFi | GPU | Framebuffer |
| h.264 accel. | pmem | Binder IPC | Compass |
| camera(s) | speakers | Accelerometer | RTC / Alarms |

# Key Challenges

- device diversity

| | | | |
|---|---|---|---|
| | | microphone | headset |
| Power | Touchscreen | Buttons | GPS |
| Cell Radio | WiFi | GPU | Framebuffer |
| h.264 accel. | pmem | Binder IPC | Compass |
| camera(s) | speakers | Accelerometer | RTC / Alarms |

- mobile usage model

# Key Challenges

- device diversity

| | | | |
|---|---|---|---|
| | | microphone | headset |
| Power | Touchscreen | Buttons | GPS |
| Cell Radio | WiFi | GPU | Framebuffer |
| h.264 accel. | pmem | Binder IPC | Compass |
| camera(s) | speakers | Accelerometer | RTC / Alarms |

- mobile usage model

➡ graphics-accelerated UI

# Key Challenges

# Cells

# Cells

# Cells

## Key Observation

# Cells

## Key Observation


large: lots of windows / apps

# Cells

## Key Observation


large: lots of windows / apps


small:
one app at a time

# Cells

## Key Observation

screen real-estate is limited, and
mobile phone users are accustomed
to interacting with *one thing* at time

# Cells

## Usage Model

# Cells

## Usage Model

*foreground*

# Cells

## Usage Model

*foreground* / *background*

# Cells

## Complete Virtualization

# Cells
## Complete Virtualization

- multiple, isolated virtual phones (VPs) on a single mobile device

# Cells
## Complete Virtualization

- multiple, isolated virtual phones (VPs) on a single mobile device
- 100% device support in each VP

# Cells
## Complete Virtualization

- multiple, isolated virtual phones (VPs) on a single mobile device
- 100% device support in each VP
  - unique phone numbers - single SIM!

# Cells
## Complete Virtualization

- multiple, isolated virtual phones (VPs) on a single mobile device
- 100% device support in each VP
  - unique phone numbers - single SIM!
  - accelerated 3D graphics!

# Cells

## Efficient Virtualization

# Cells
## Efficient Virtualization

- less than 2% overhead in runtime tests

# Cells
## Efficient Virtualization

- less than 2% overhead in runtime tests
- imperceptible switch time among VPs

# Cells

## Transparent Virtualization

# Cells
## Transparent Virtualization

- each VP sees / uses all devices

# Cells
## Transparent Virtualization

- each VP sees / uses all devices
- user can run any unmodified apps

# Cells
## Transparent Virtualization

- each VP sees / uses all devices
- user can run any unmodified apps
- foreground VP switches like an app

# Single Kernel: Multiple VPs

Linux
Kernel

# Single Kernel: Multiple VPs


Linux
Kernel

# Single Kernel: Multiple VPs

VP 1 VP 2 VP 3 ...

virtualize at OS interface

Linux Kernel

# Single Kernel: Multiple VPs

isolated collection of processes

virtualize at OS interface


Linux Kernel

# Single Kernel: Device Support


Linux
Kernel

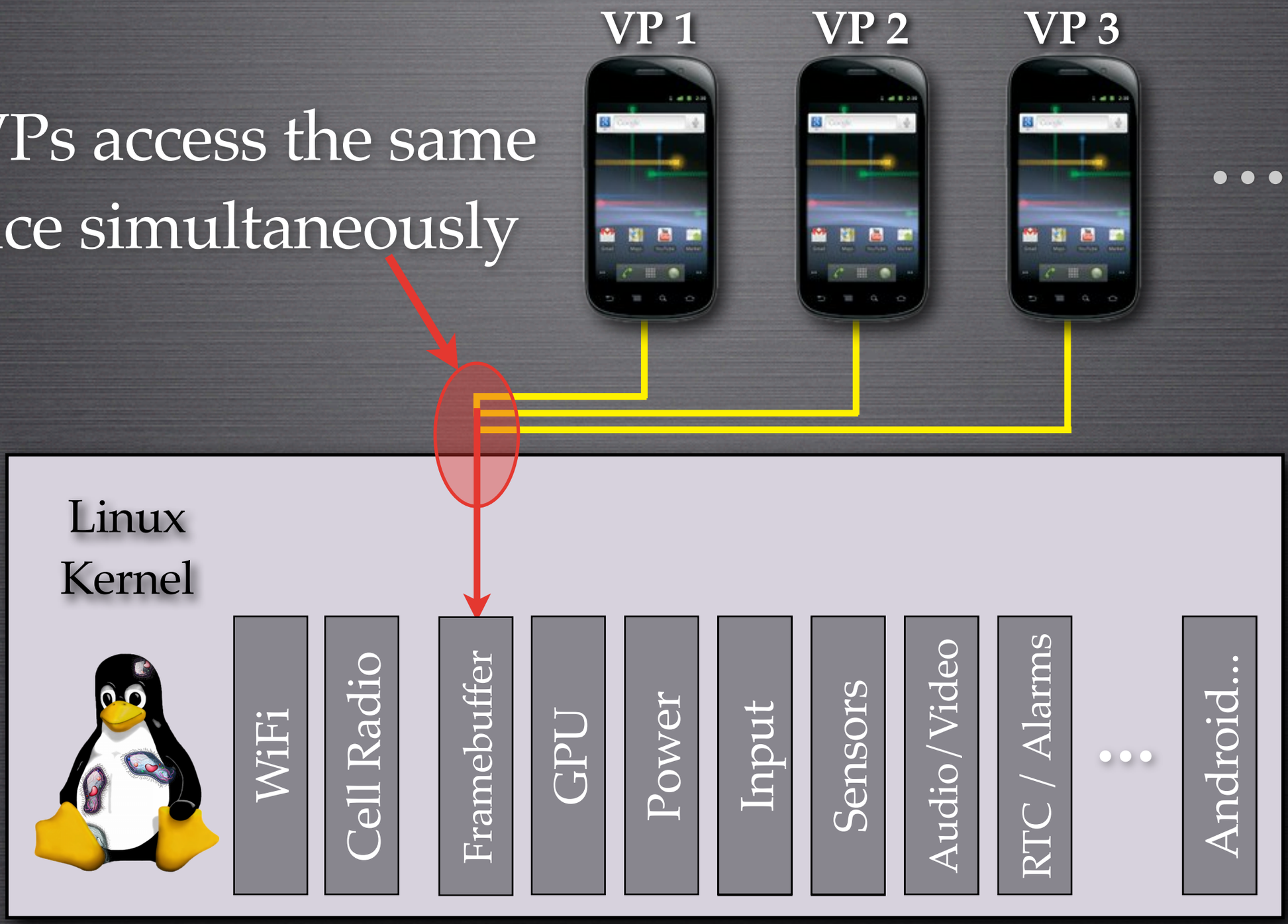
# Single Kernel: Device Support



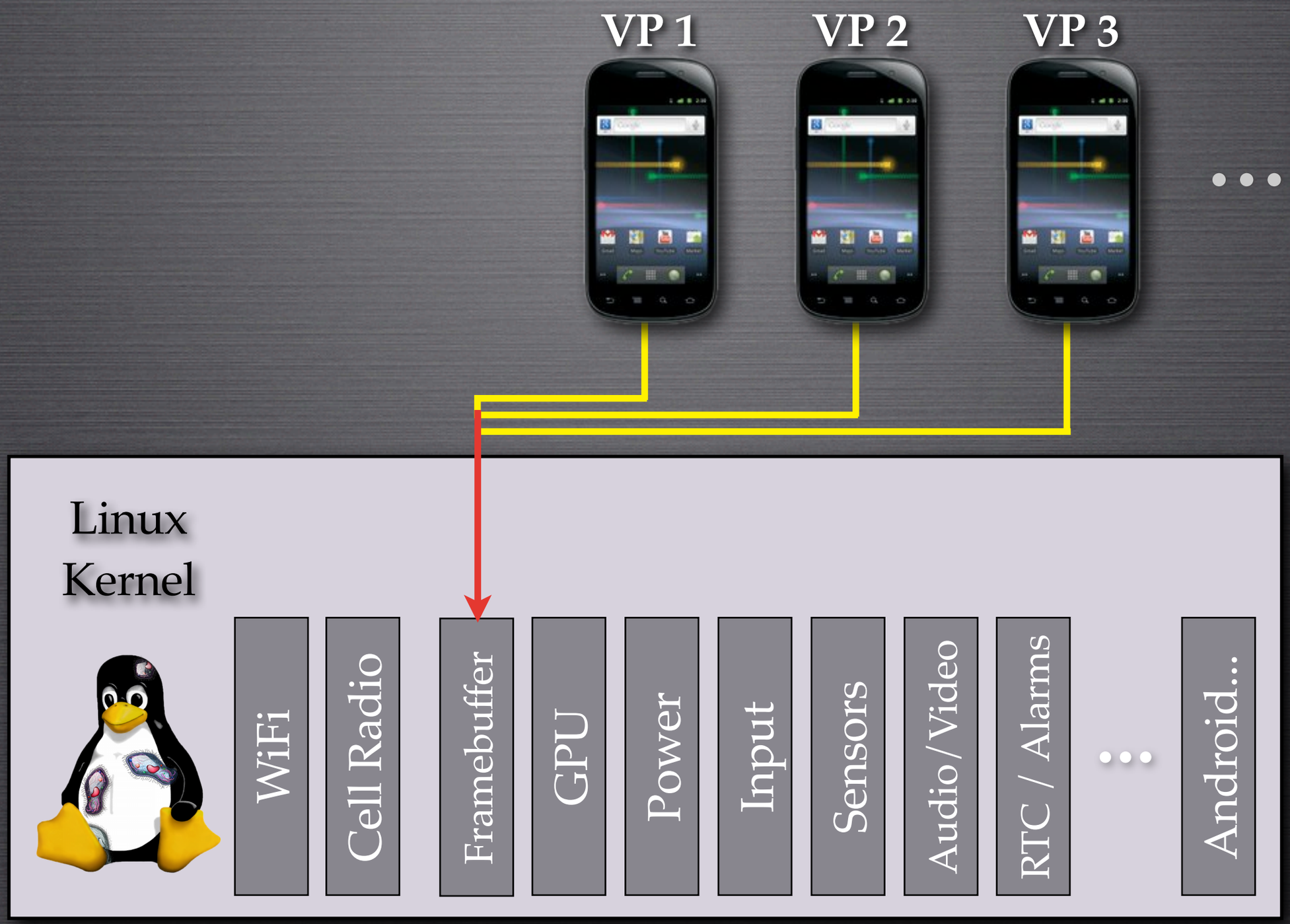
 

# Single Kernel: Device Support

all VPs access the same device simultaneously

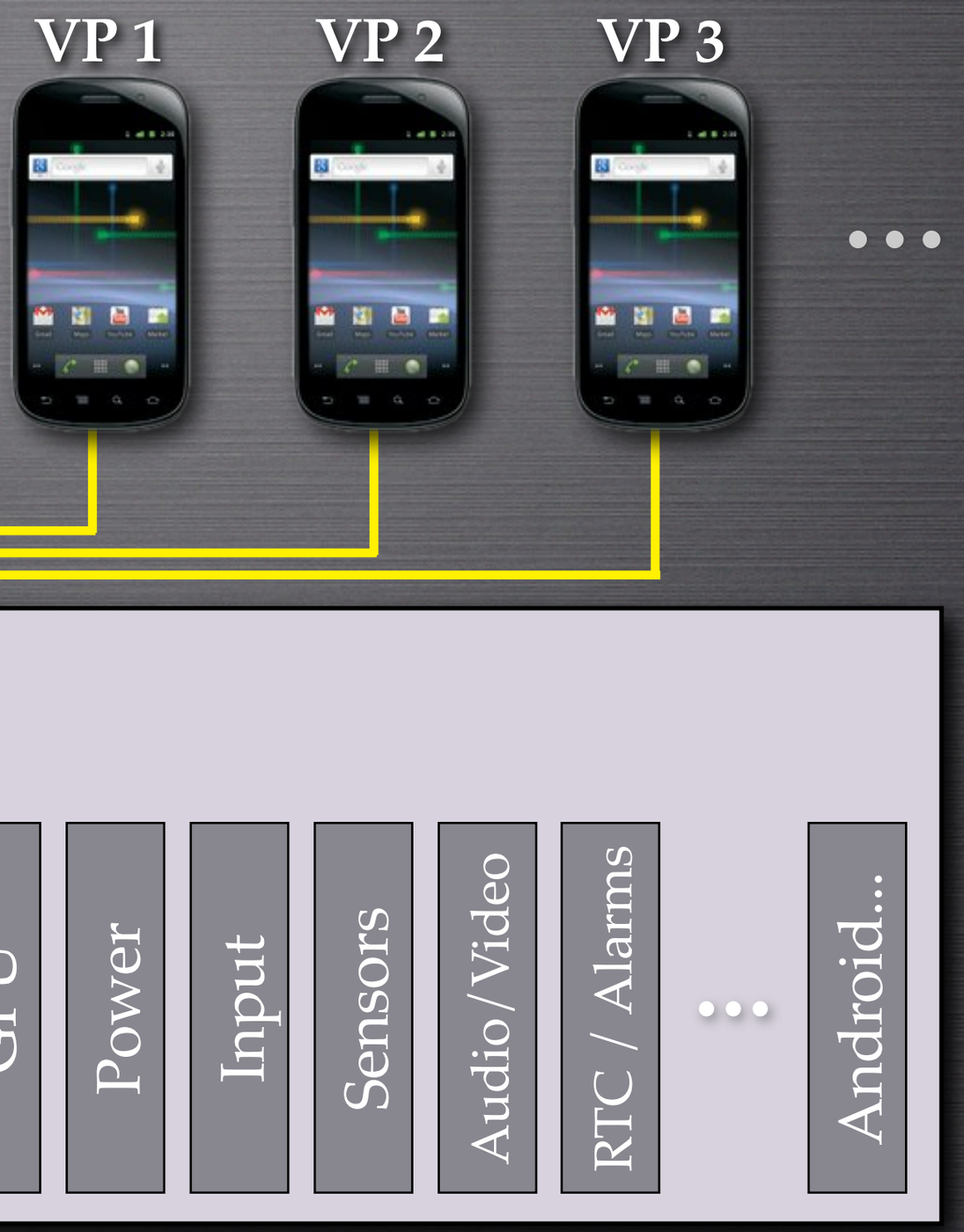




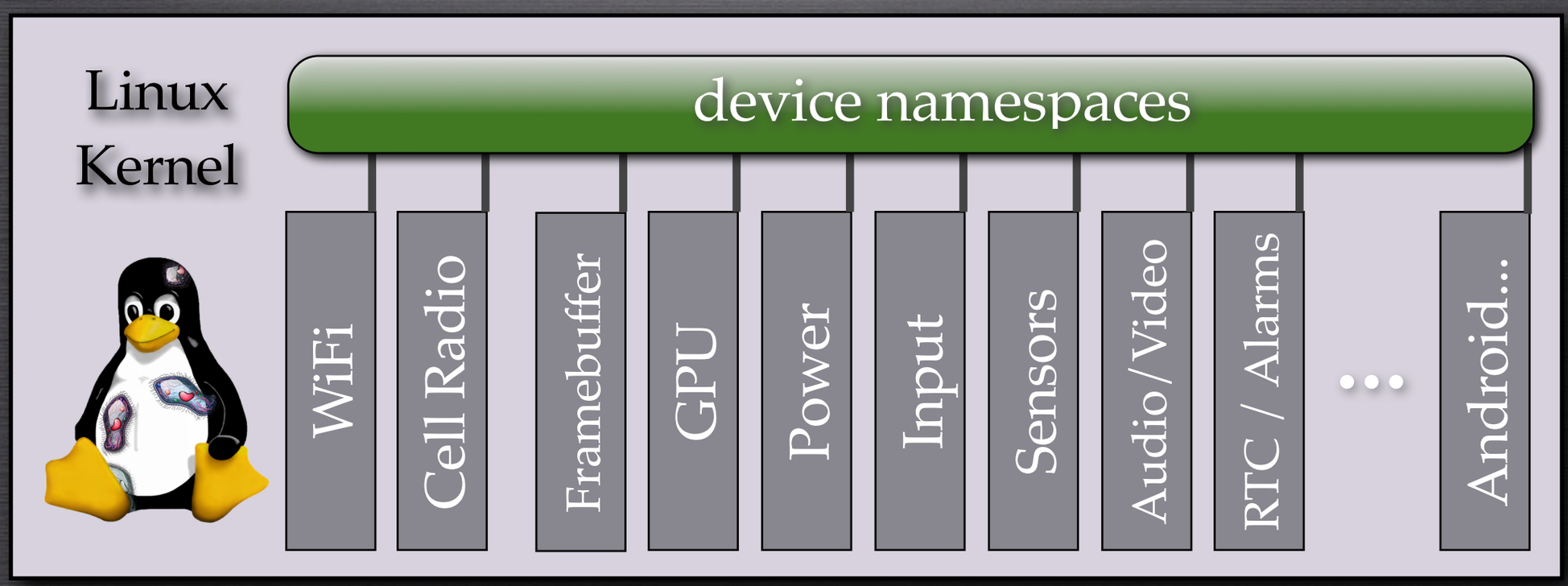
*safely, correctly multiplex access to devices*



# Device Namespaces

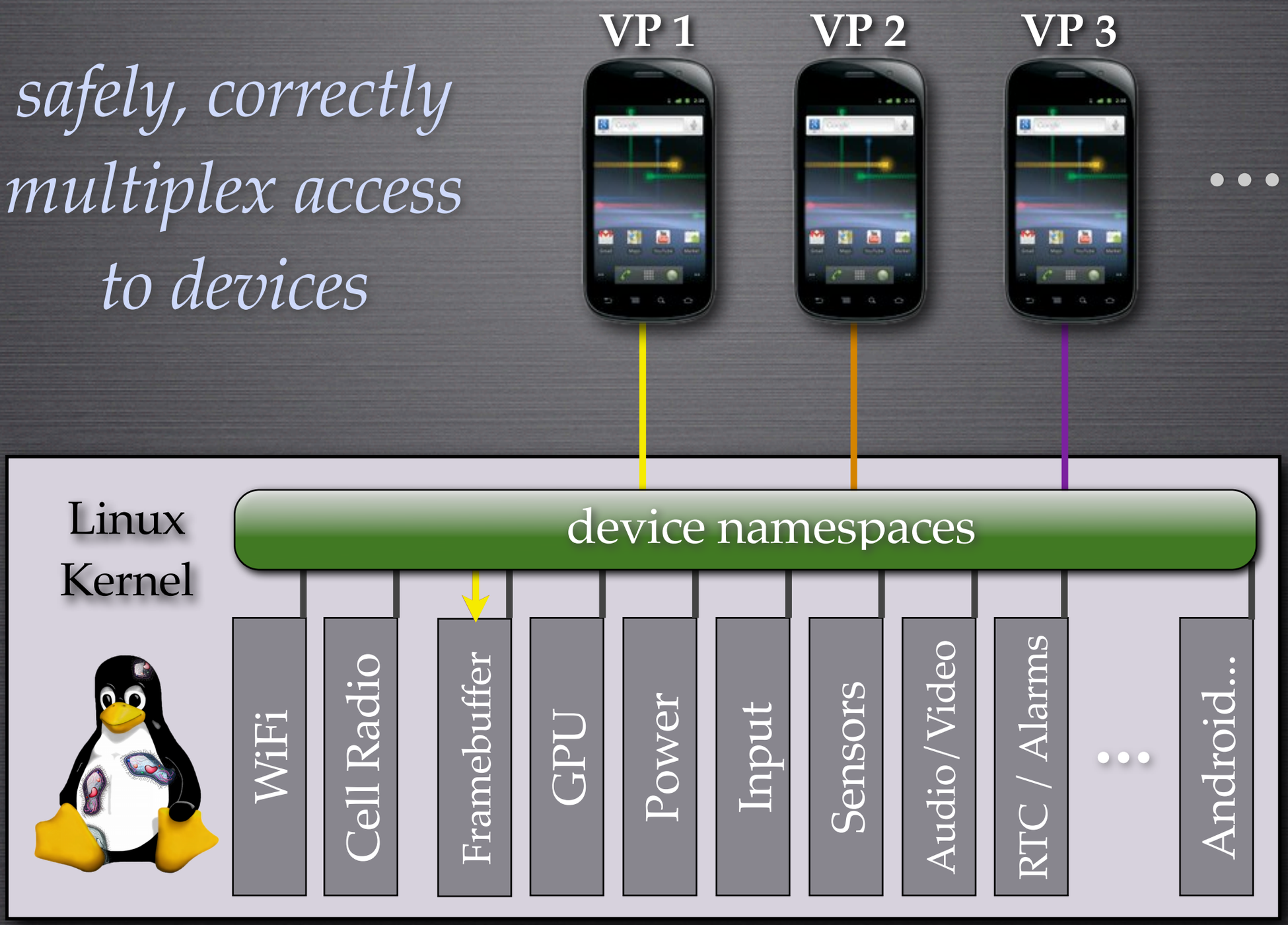
*safely, correctly multiplex access to devices*

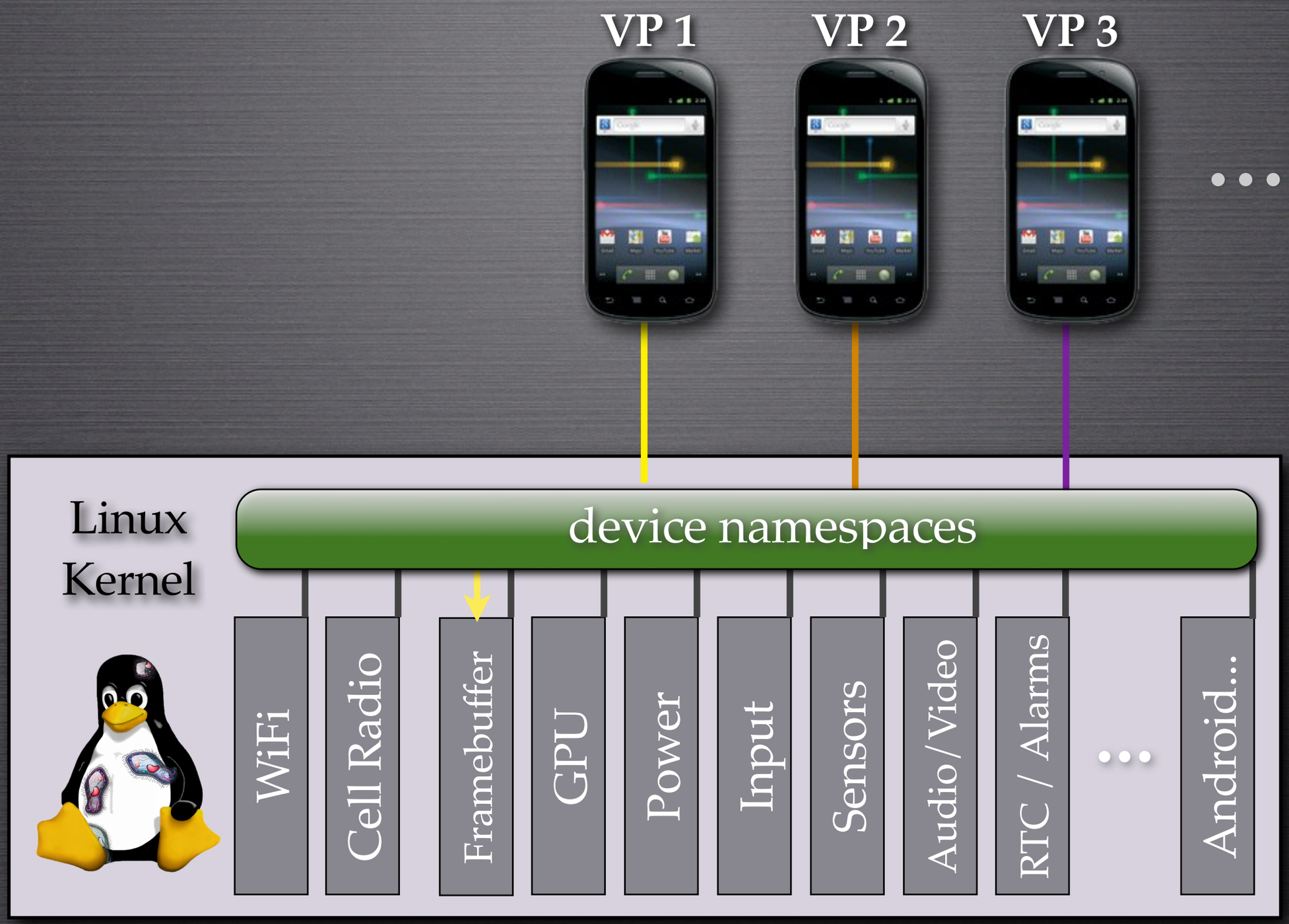
# Device Namespaces

*safely, correctly multiplex access to devices*



# Cells



# Cells

*device namespaces*

+

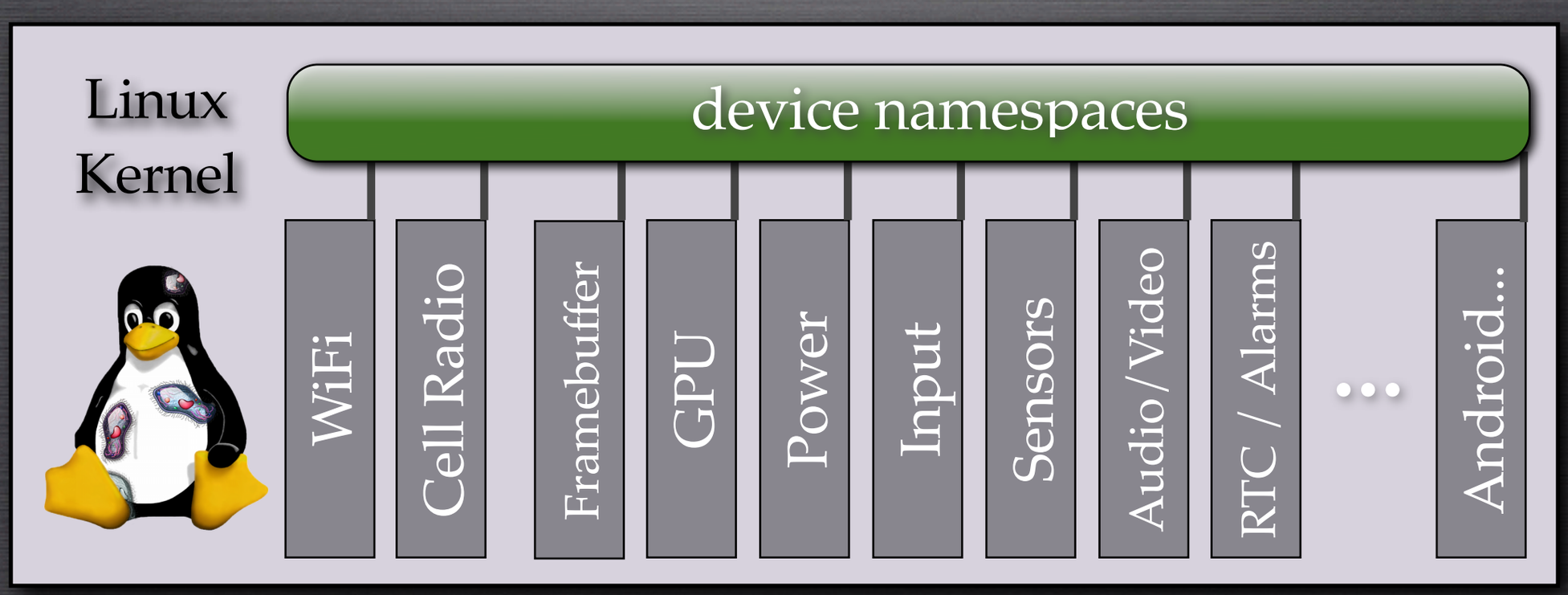
*foreground* / *background*

=

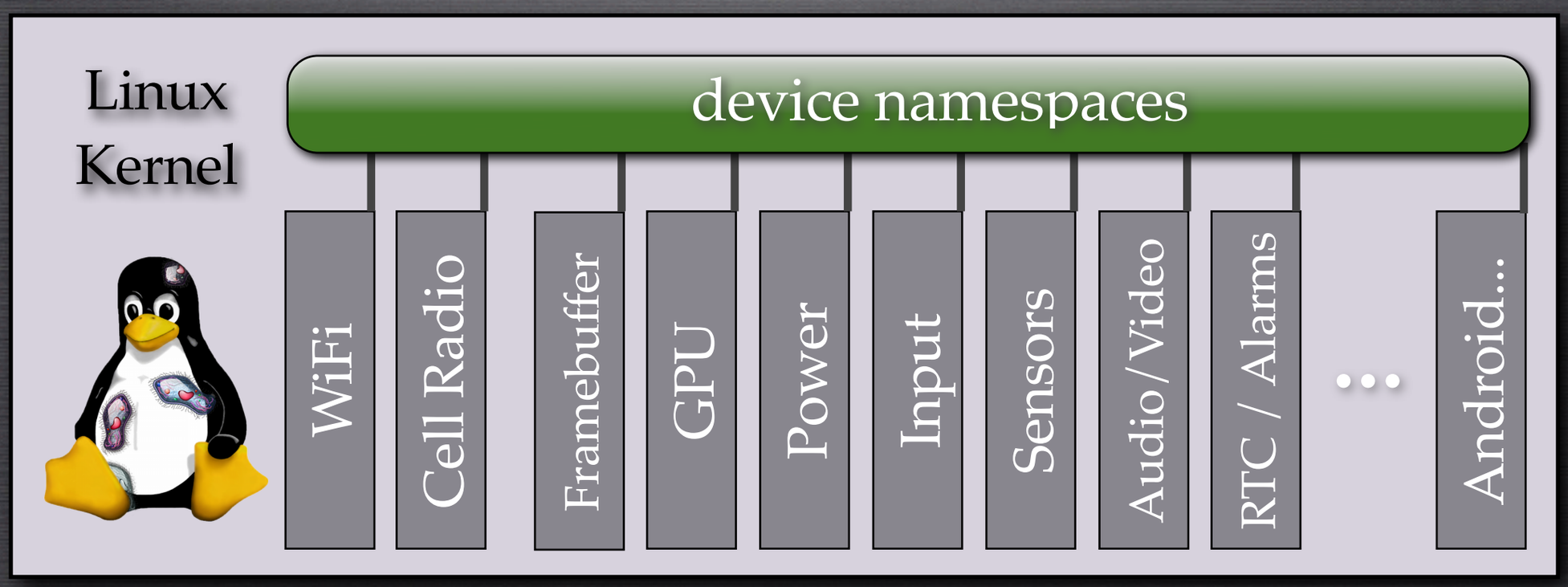
## Complete, Efficient, Transparent Mobile Virtualization

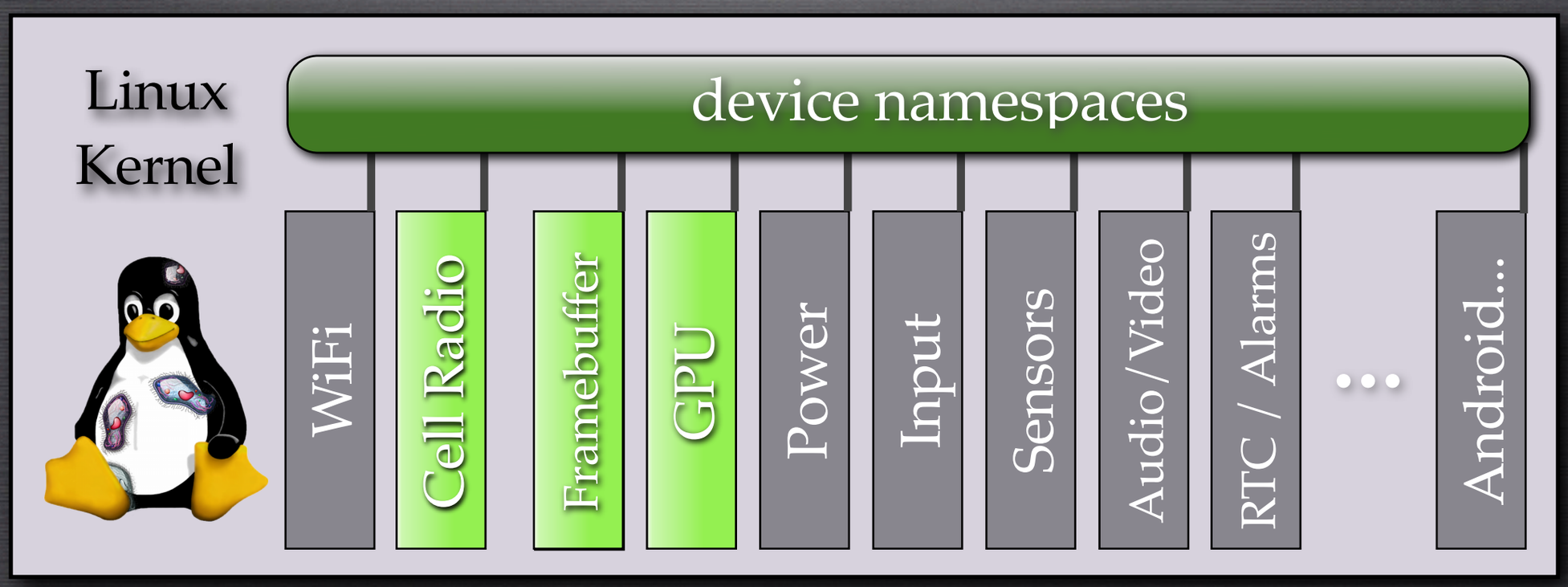
# Cells







Framebuffer

GPU

Cell Radio

Framebuffer

GPU

Cell Radio

efficient basic graphics virtualization

Framebuffer — efficient basic graphics virtualization

GPU — hardware accelerated graphics

Cell Radio

| | |
|---|---|
| Framebuffer | efficient basic graphics virtualization |
| GPU | hardware accelerated graphics |
| Cell Radio | proprietary / closed interface |

Framebuffer

GPU


Framebuffer


screen memory


Framebuffer

virtual addresses

physical addresses

screen memory


Framebuffer

# Approach 1: Single Assignment

# Approach 2: Emulated Hardware

screen memory


Framebuffer

# Approach 2: Emulated Hardware

screen memory


Framebuffer

# Approach 2: Emulated Hardware

emulated framebuffer

Framebuffer

screen memory

# Approach 2: Emulated Hardware

emulated framebuffer

screen memory


Framebuffer

virtual state

# Approach 2: Emulated Hardware

emulated framebuffer

screen memory


Framebuffer

virtual state

# Approach 2: Emulated Hardware

emulated framebuffer

screen memory


Framebuffer

virtual state

Framebuffer

# Cells: Device Namespaces

VP 1 VP 2 VP 3

screen memory


Framebuffer

# Cells: Device Namespaces

mux_fb presents identical device interface to all VPs using *device namespaces*

VP 1

VP 2

VP 3

mux_fb

screen memory


Framebuffer

# Cells: Device Namespaces

VP 1

foreground

VP 2

background

VP 3

background

virtual addresses

physical addresses

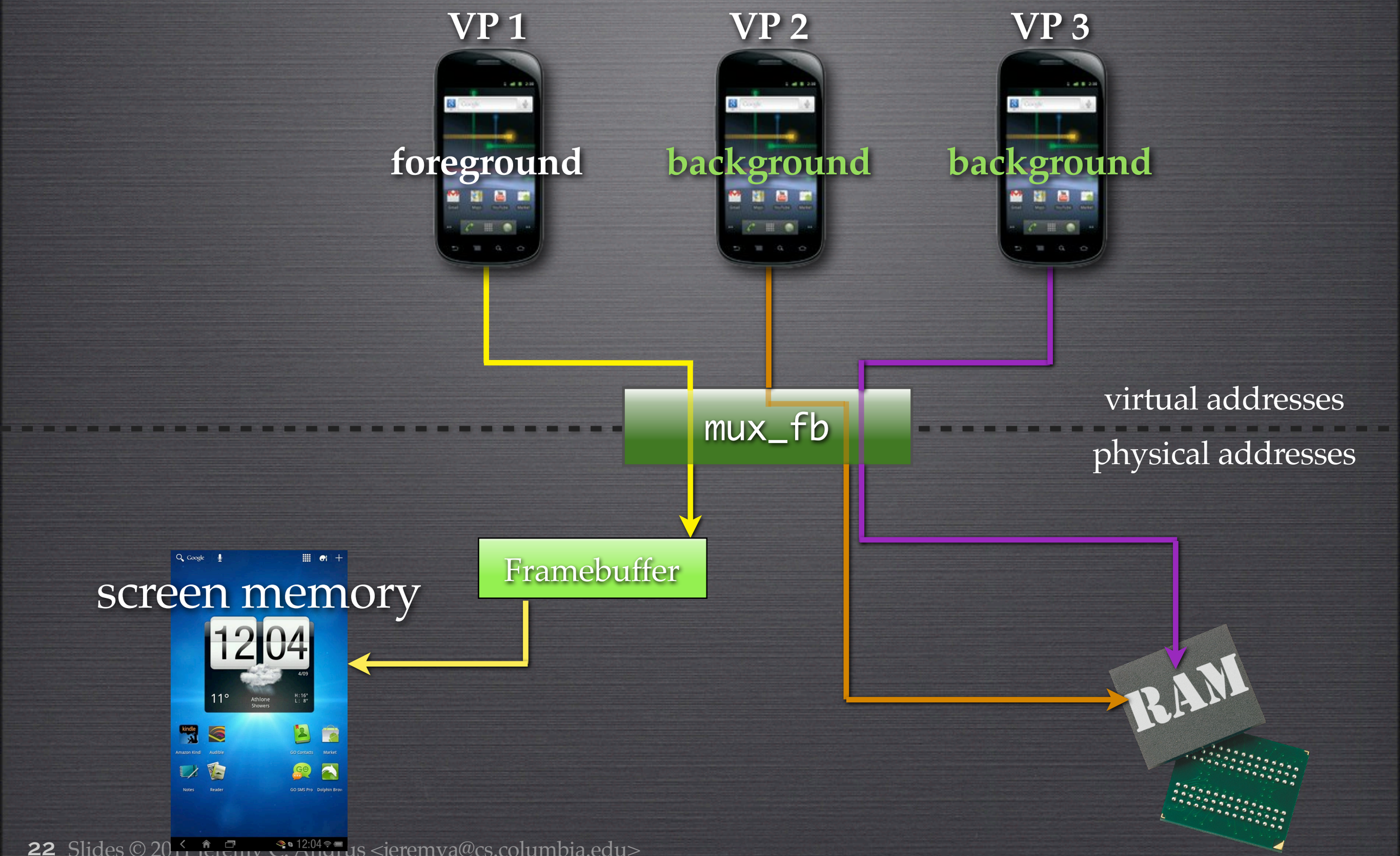
mux_fb

Framebuffer

screen memory

# Cells: Device Namespaces



# Cells: Device Namespaces

VP 1

background

VP 2

foreground

VP 3

background

swap virt addr mappings:
point to different phys addr

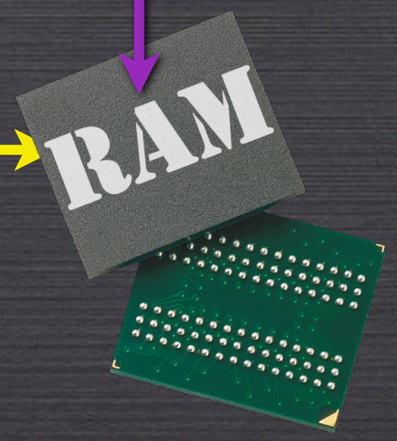
mux_fb

virtual addresses

physical addresses

Framebuffer

screen memory

# Accelerated Graphics

VP 1 VP 2 VP 3


screen memory

Framebuffer

GPU

# Accelerated Graphics

VP 1 VP 2 VP 3


screen memory

GPU

# Accelerated Graphics



# Accelerated Graphics




OpenGL context

OpenGL context

OpenGL context

process isolation


MMU

GPU

graphics virtual addresses

physical addresses

# Accelerated Graphics

VP: just a set of processes!


process isolation

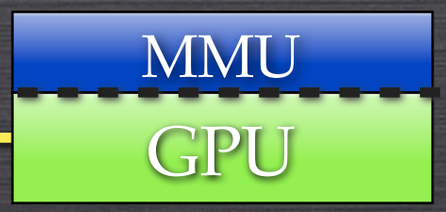


graphics virtual addresses

physical addresses





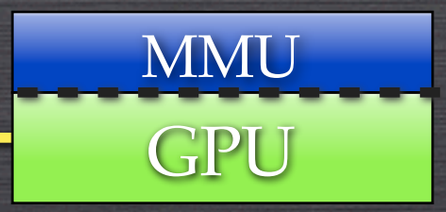

graphics virtual addresses

physical addresses

# Device Namespace + Graphics Context






graphics virtual addresses

physical addresses

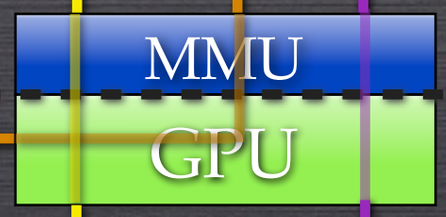
# Device Namespace + Graphics Context



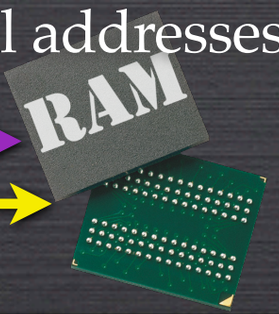
# Device Namespace + Graphics Context






graphics virtual addresses

physical addresses



Baseband: GSM / CDMA

Vendor RIL

Drivers

Linux Kernel

Baseband: GSM / CDMA

RilD

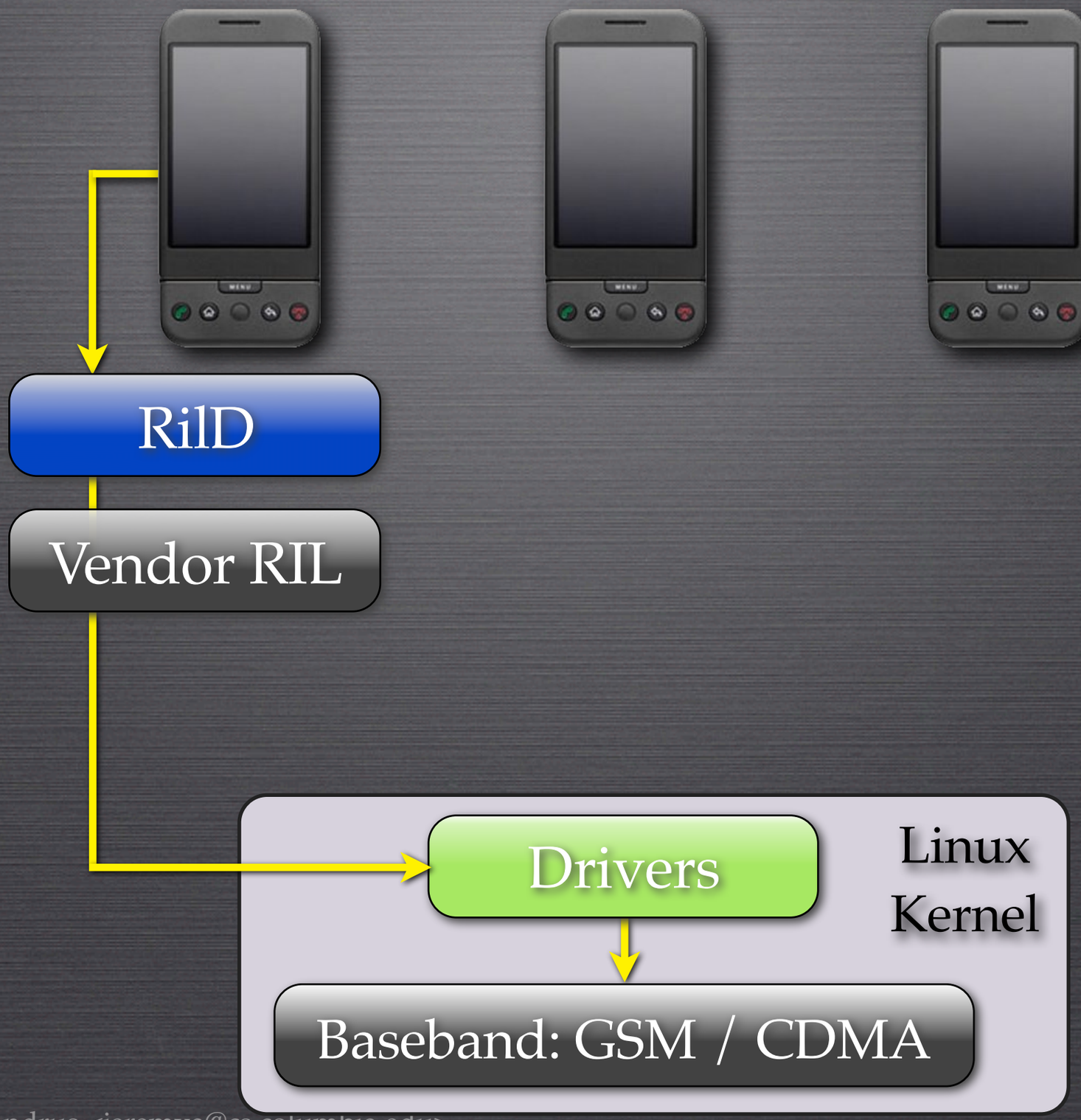
Vendor RIL

Linux Kernel

Drivers

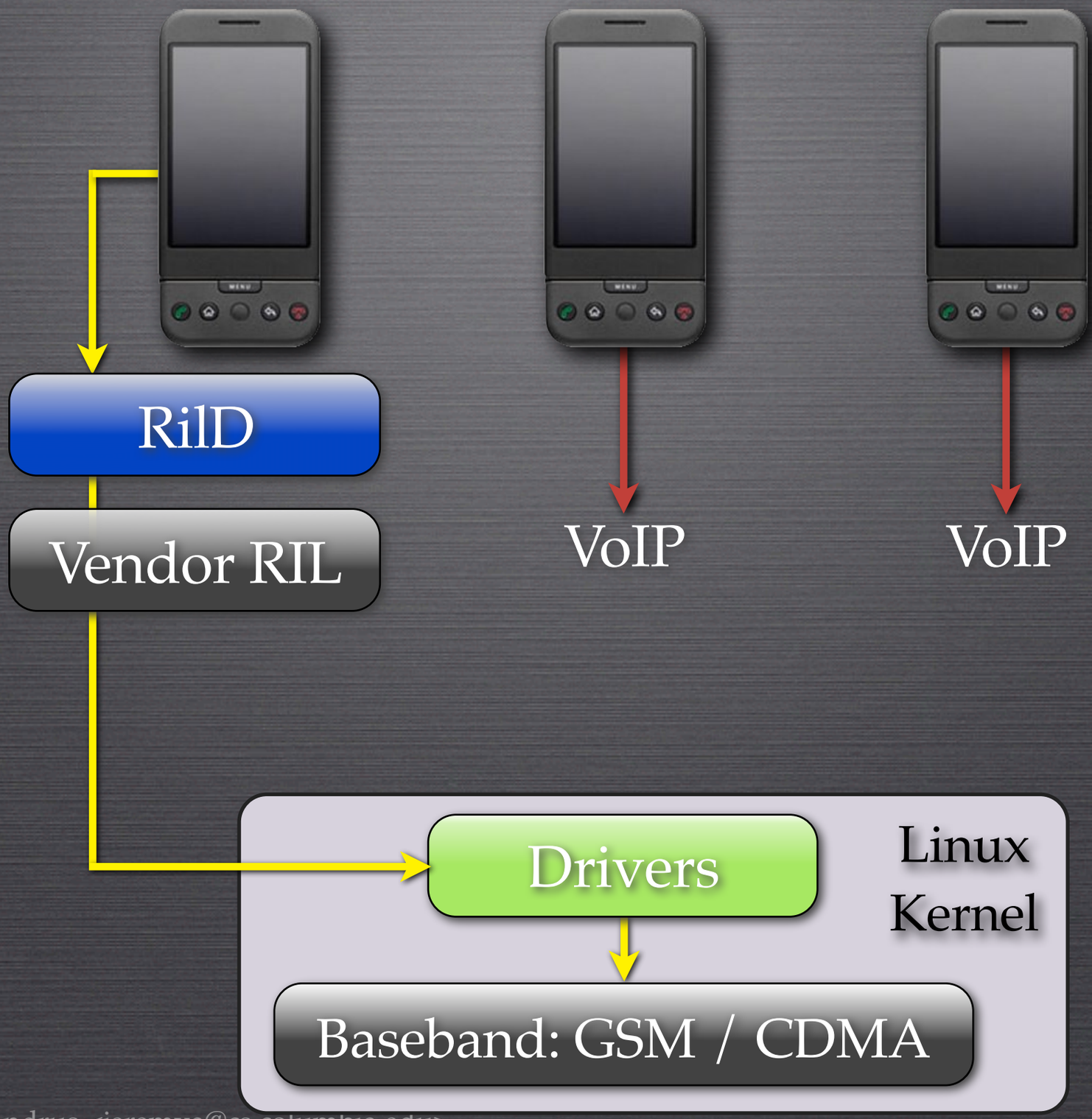
Baseband: GSM / CDMA



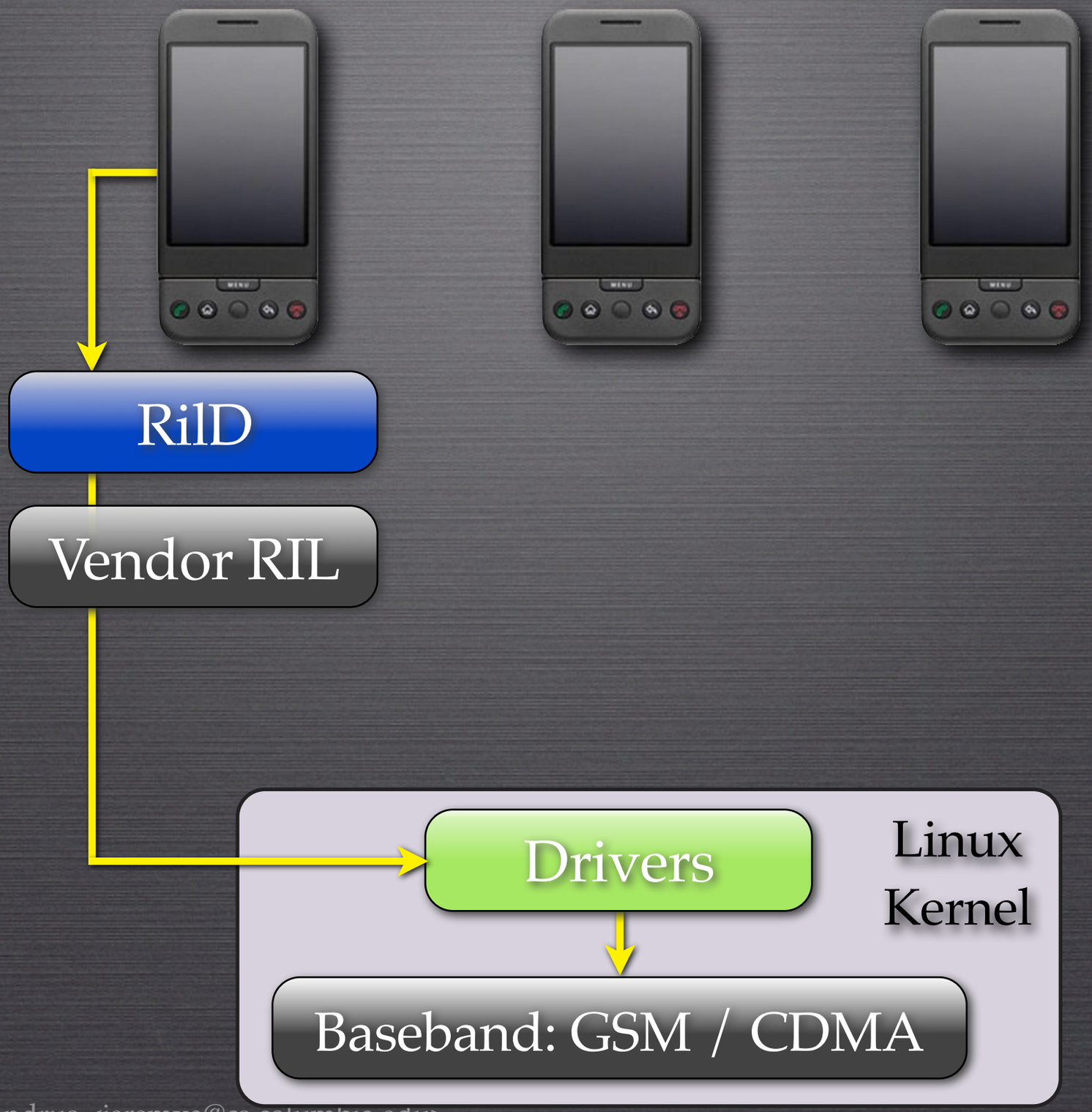

 

# VoIP?



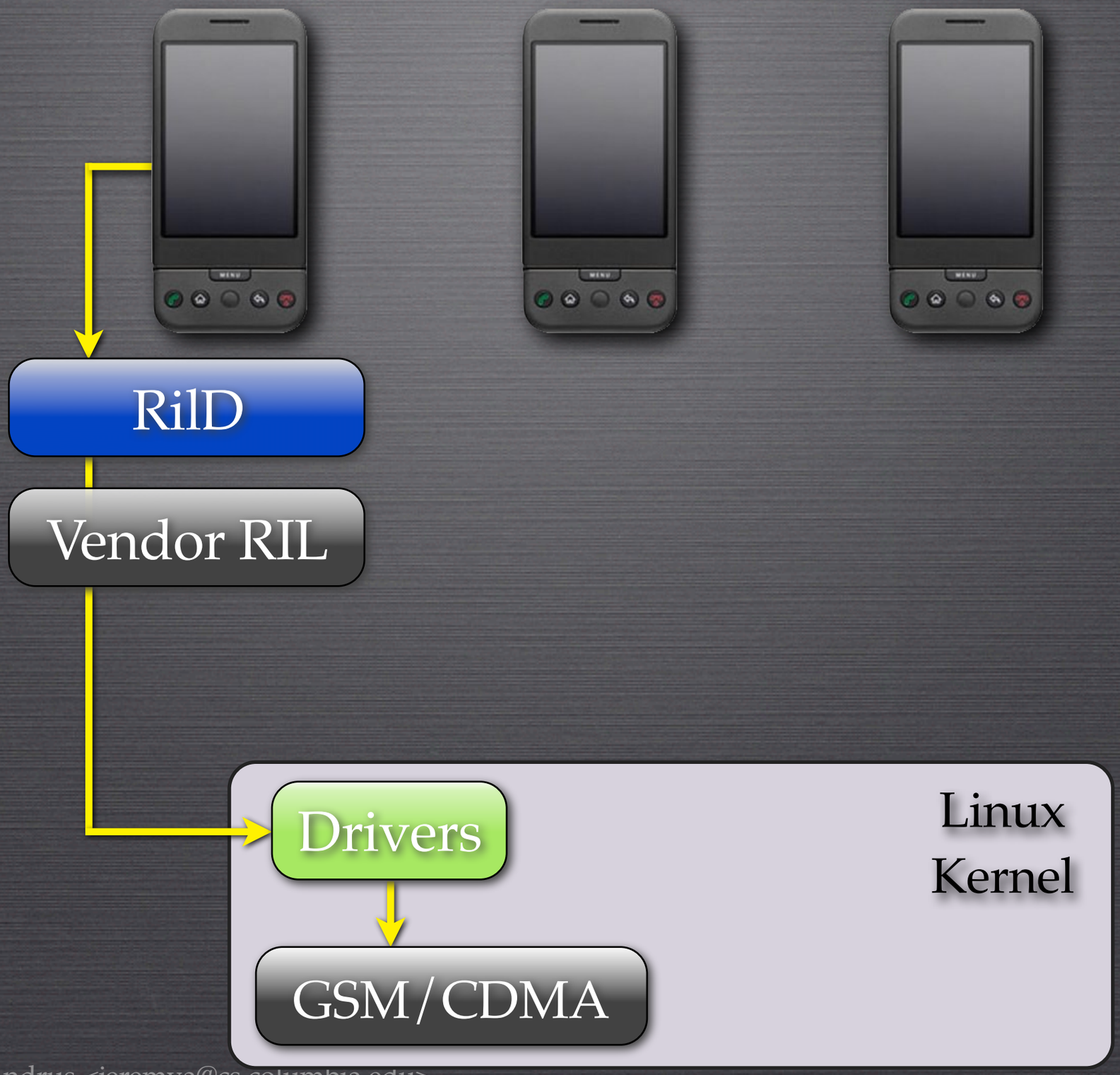

 

# VoIP?



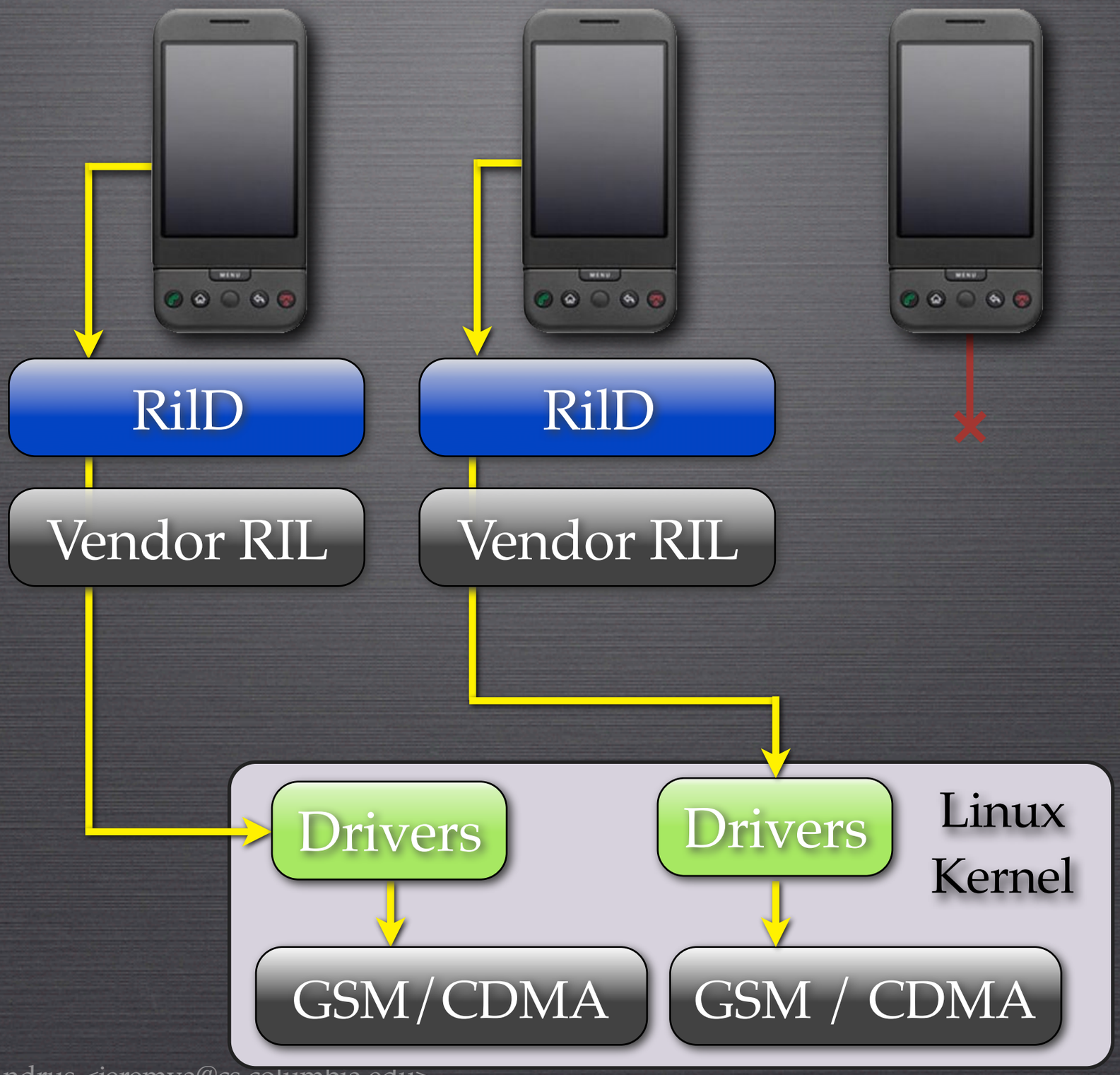

 

# Dual-SIM?



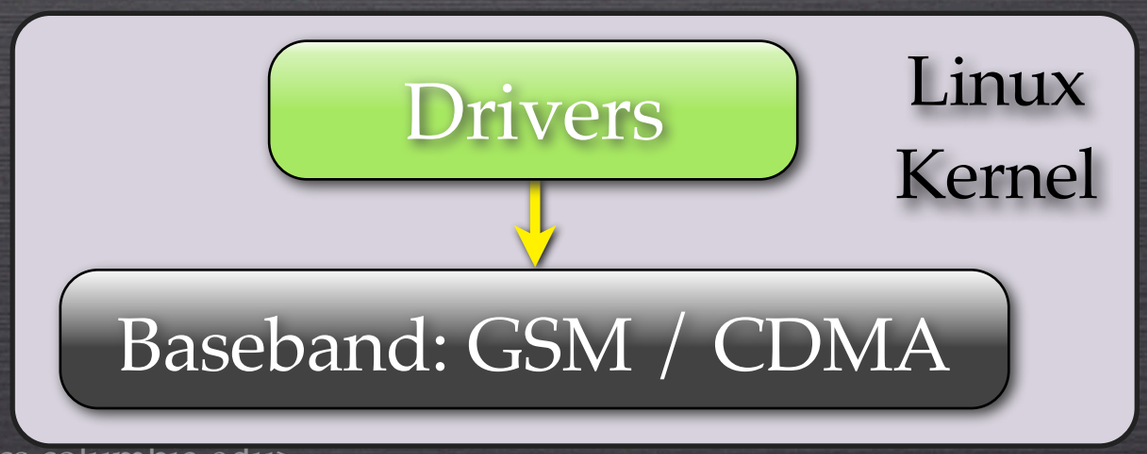
# Dual-SIM?



RilD

Vendor RIL



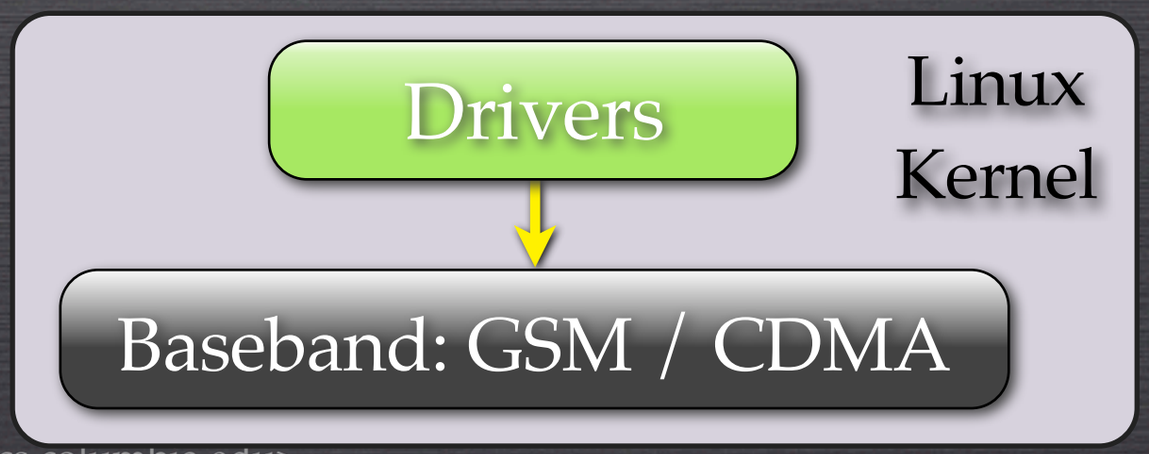
VP 1

VP 2

VP 3

RilD

Vendor RIL



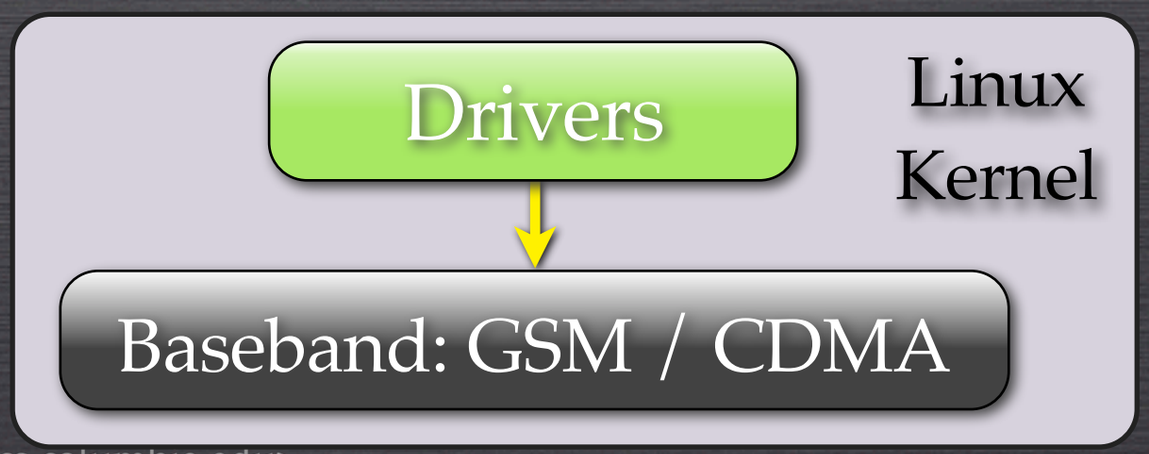

 


RilD

proprietary hardware/software

Vendor RIL

requires a well-defined interface.



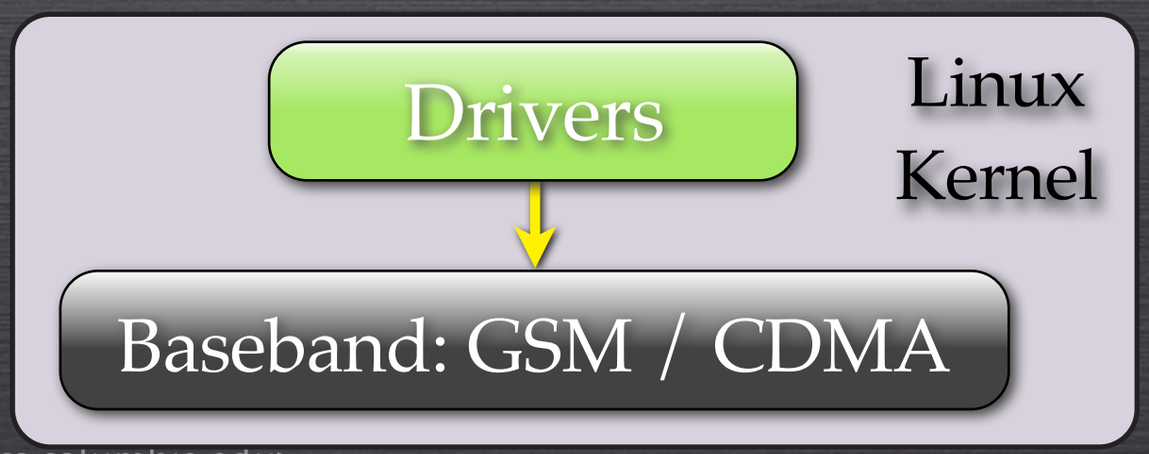

 


vendor API

RilD

proprietary hardware / software requires a well-defined interface.

Vendor RIL



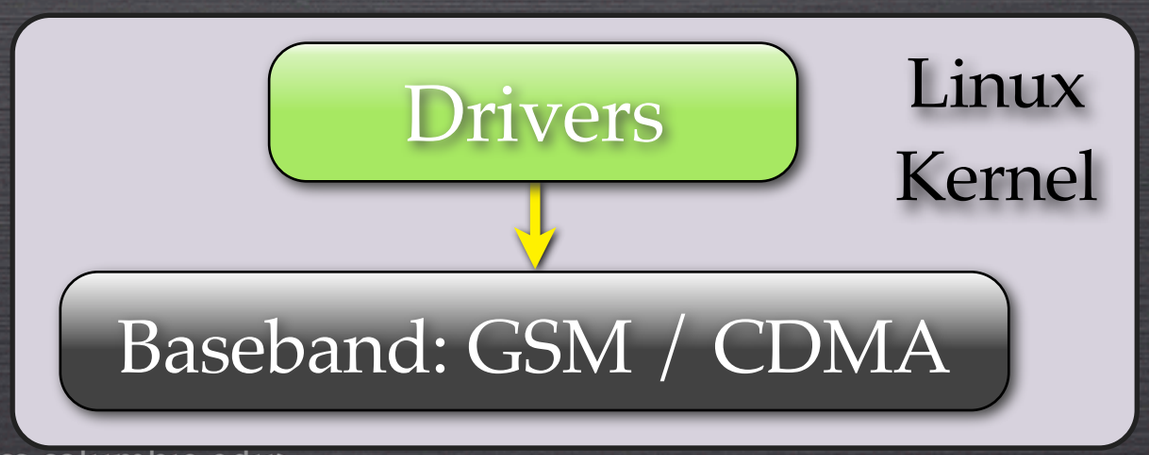

 


vendor API

RilD

proprietary hardware / software requires a well-defined interface.

Vendor RIL



# Cells: User-Level Namespace Proxy

# Cells: User-Level Namespace Proxy


RilD

vendor API

Cells RIL

Vendor RIL

Root Namespace

Drivers

Linux Kernel

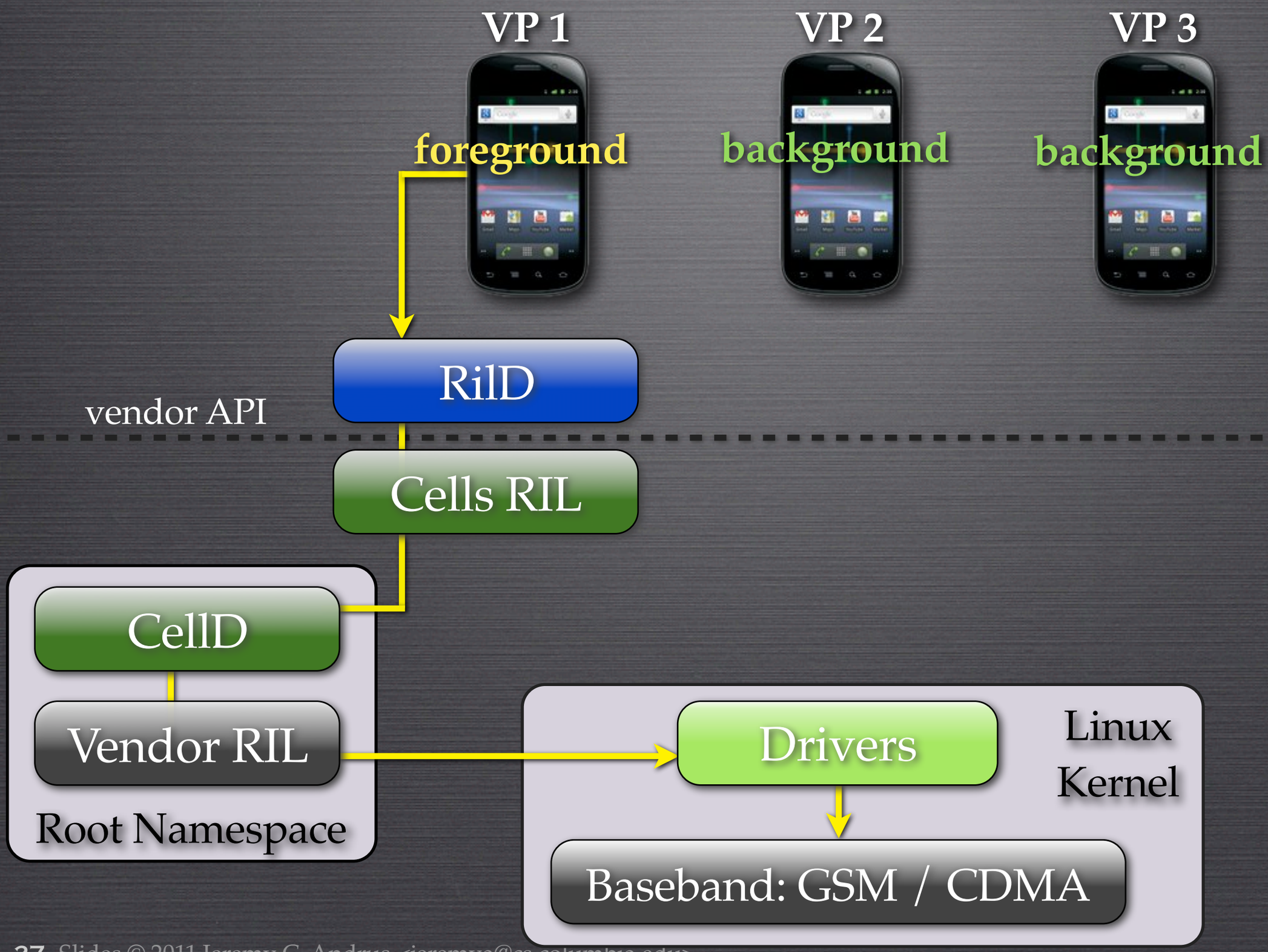
Baseband: GSM / CDMA

# Cells: User-Level Namespace Proxy

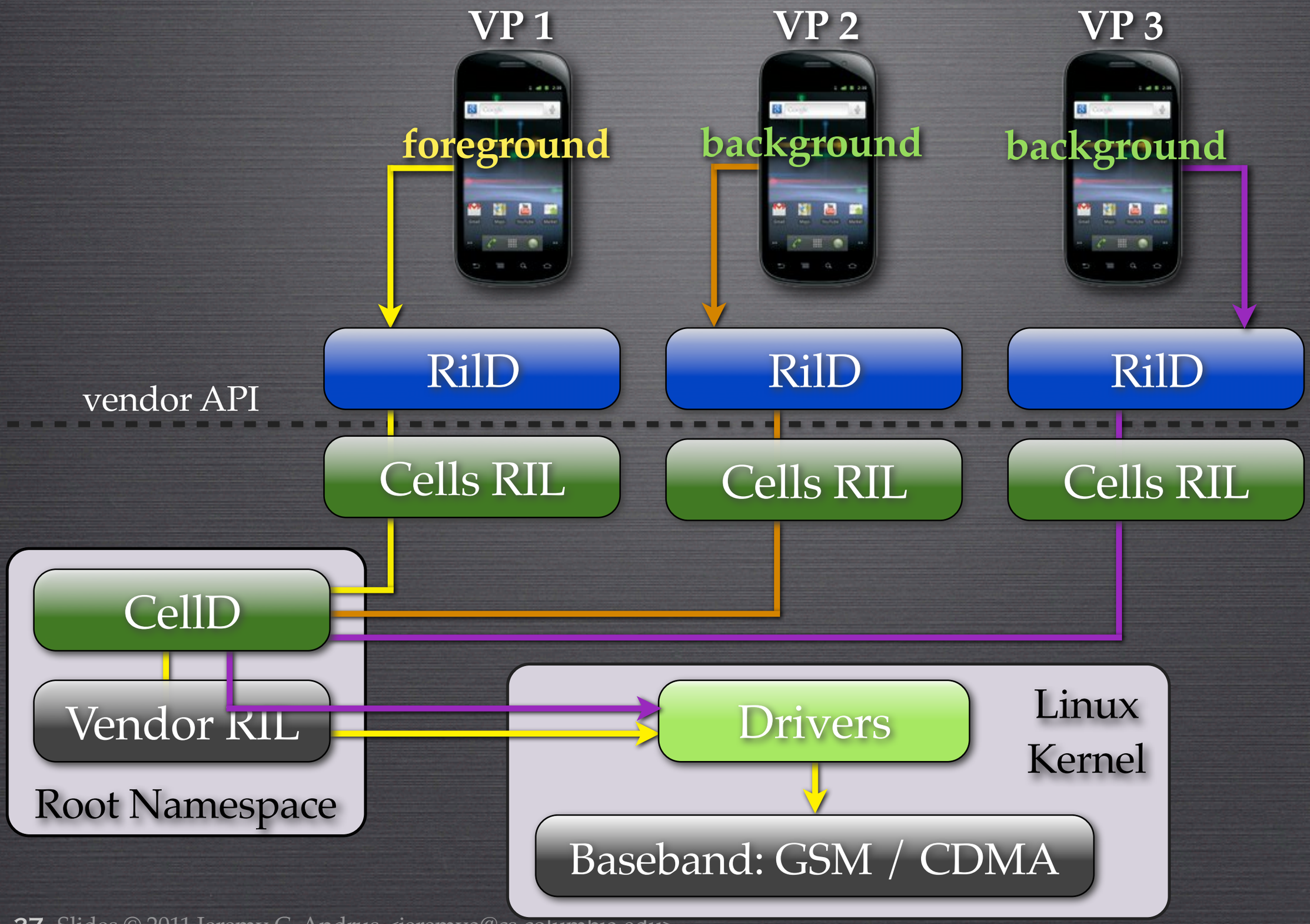

 

# Cells: User-Level Namespace Proxy



# Cells: User-Level Namespace Proxy



# Experimental Results

## Setup

# Experimental Results

## Setup

- Nexus S

# Experimental Results

## Setup

- Nexus S
- *five* virtual phones

# Experimental Results

## Setup

- Nexus S
- *five* virtual phones
- overhead vs. stock ⟨*Android 2.3*⟩

# Experimental Results

## Setup

# Experimental Results
## Setup

- CPU ⟨*Linpack*⟩

# Experimental Results

## Setup

- CPU ⟨*Linpack*⟩
- graphics ⟨*Neocore*⟩

# Experimental Results

## Setup

- CPU ⟨*Linpack*⟩
- graphics ⟨*Neocore*⟩
- storage ⟨*Quadrant*⟩

# Experimental Results

## Setup

- CPU ⟨*Linpack*⟩
- graphics ⟨*Neocore*⟩
- storage ⟨*Quadrant*⟩
- web browsing ⟨*Sun Spider*⟩

# Experimental Results

## Setup

- CPU ⟨*Linpack*⟩
- graphics ⟨*Neocore*⟩
- storage ⟨*Quadrant*⟩
- web browsing ⟨*Sun Spider*⟩
- networking ⟨*Custom WiFi Test*⟩

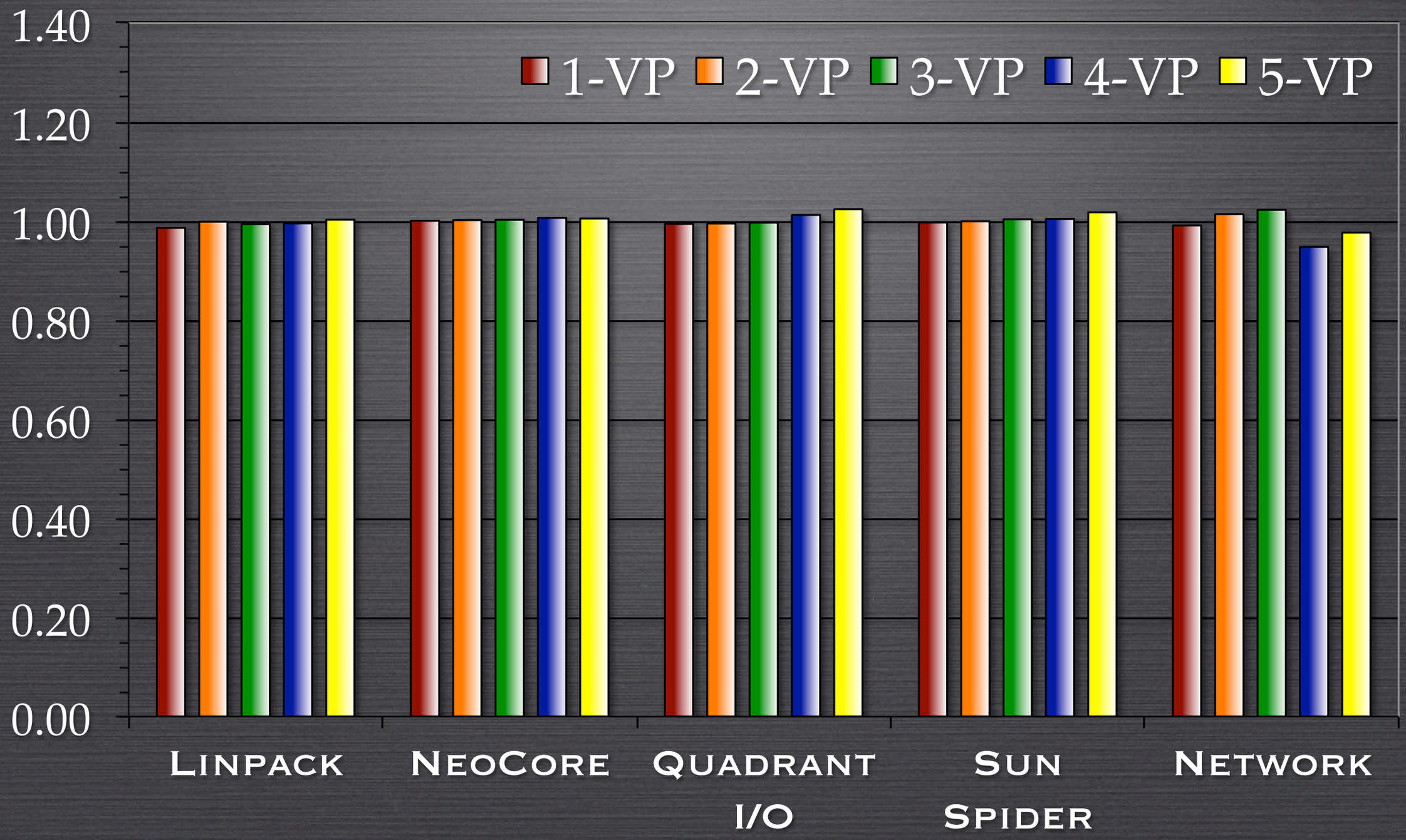
# Experimental Results

## Runtime Overhead

# Experimental Results

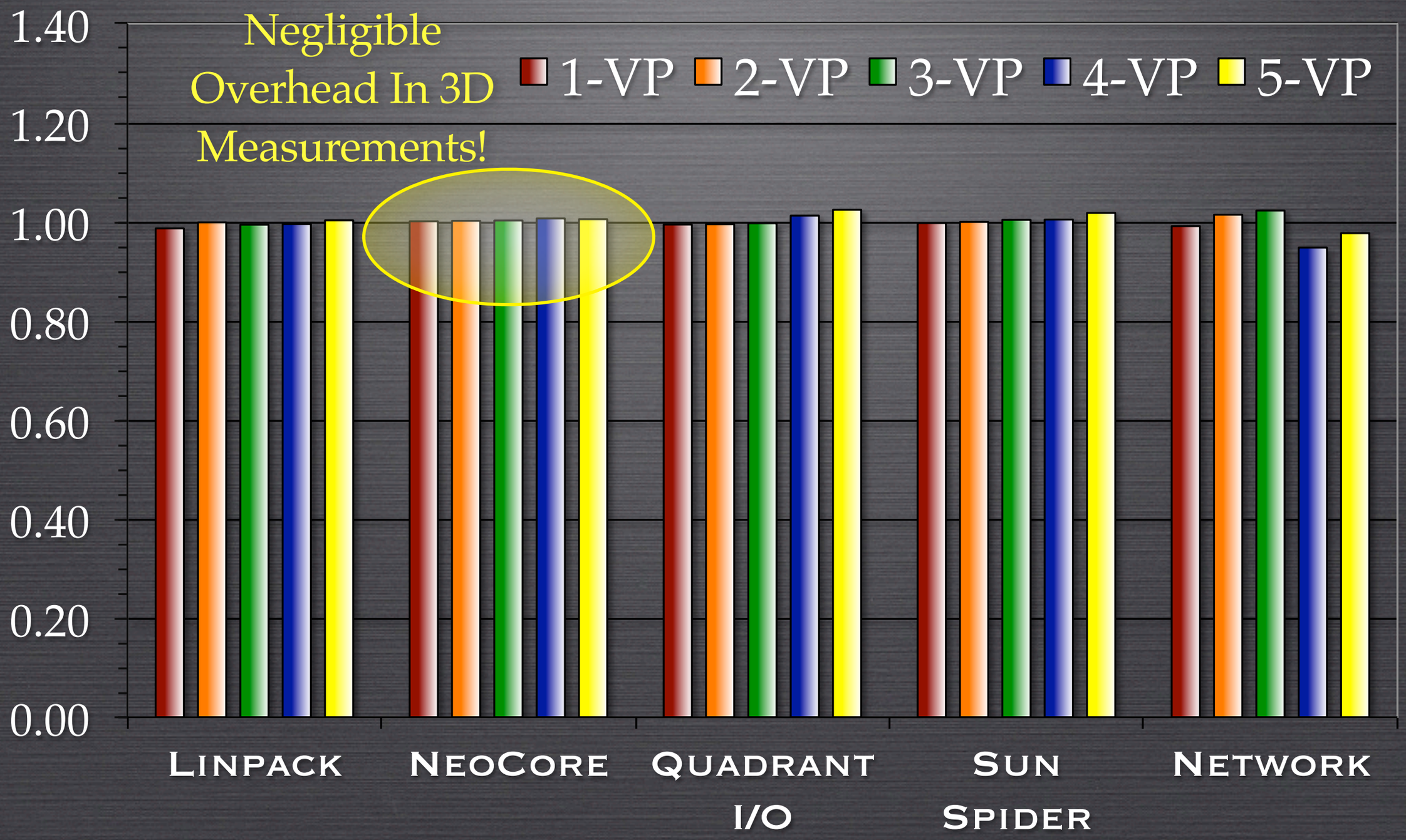
## Runtime Overhead



# Experimental Results

## Runtime Overhead



# Demo

# Cells

## Complete, Efficient, Transparent Mobile Virtualization

# Cells

## Complete, Efficient, Transparent Mobile Virtualization

- device namespaces
  - safely and efficiently share devices

# Cells

## Complete, Efficient, Transparent Mobile Virtualization

- device namespaces
  - safely and efficiently share devices
- foreground / background
  - designed specifically for mobile devices

# Cells

## Complete, Efficient, Transparent Mobile Virtualization

- device namespaces
  - safely and efficiently share devices
- foreground / background
  - designed specifically for mobile devices
- implemented on Android

# Cells

## Complete, Efficient, Transparent Mobile Virtualization

- device namespaces
  - safely and efficiently share devices
- foreground / background
  - designed specifically for mobile devices
- implemented on Android
- less than 2% overhead on Nexus S

# More Info

Software Systems Laboratory

cells.cs.columbia.edu

CELLROX

cellrox.com